UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/803,514	03/08/2001	Sridhar Obilisetty	026507-000300US	6110

20350 7590 12/13/2007
TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
12/13/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 09/803,514	**Applicant(s)** OBILISETTY, SRIDHAR	
	Examiner Tuan A. Vu	**Art Unit** 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 November 2007*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-6 and 8-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6, 8-25* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 11/01/07.

As indicated in Applicant's response, claims 1, 13, 25 have been amended, and claims 7, 26-36 canceled. Claims 1-6, 8-25 are pending in the office action.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-6, 8-12, and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bloch et al., USPN: 2002/0129129 (hereinafter Bloch).

As per claim 1, Bloch discloses a method for implementing an application on a client computer system, said method comprising:

receiving at said client a plurality of text files; each defining a component of the application (e.g. Fig. 1; steps 66, 68, 70 - Fig. 5; Fig. 6; pg. 10-11, para 0095, 0096);

executing a program resident on said client system for using a combination of said text files to create an application (e.g. *AVM 221* - Fig. 2; para 0068-0069, pg. 8; *GUI frame ... waits for the user* - para 0057-0058, pg. 6 – Note: installed and iniatialized AVM -- para 0037, pg. 4-- providing GUI rendered components for user events to take place reads on executing program resident on client); and

creating said application on said client system according to said program (e.g. Fig. 2, 5), wherein said application is executable independent from said program (e.g. *Client Tasks, Host* *tasks* – para 0086-87, pg. 10; *once ... initialized ... user interact ...visible frame ... next user* interaction ... *System handler 315 to exit* - para 00100, pg. 11; *message box ... tasks ... unique to the operating system* – para 0102, pg. 11; *remote procedures* – para 0103, pg. 11; *Database handler ... user to access data ... Database* – para 0105, pg. 12; *user interface ... accessing any database ... to test new software* – para 0109 – Note: user-driven actions - para 0057-0058, pg. 6 - to invoke methods, system tasks or script based remote calls or for testing of components **read on** executables of assembled applications within the AVM non-application-specific framework – see para 0037, pg. 4(emphasis added) – and/or execution thereof that depend on user, i.e. **contextually independent** from the Gui layer or interface support by the launched AVM).

But Bloch does not explicitly disclose checking automatically for updated versions of said text files. Bloch however addresses the urge for providing latest set of files in accordance to appropriate version of script file or virtual machine files with regard to version (e.g. para 0051-0053, pg. 5-6); hence has taught checking of files and their latest upgrade; and further discloses parsing XML or HTTP text files and browser (Fig. 5; para 0030, pg. 3). It is noted that a file version being included in the header of XML or HTTP files is implicitly disclosed owing to known browser/markup language technologies at the time the invention was made. In light of the desirability of updating browser files to meet the appropriate virtual machine or execution environment version as mentioned above along with the implicitly automated version/format compliance checking when markup files are processed by a browser engine, it would have been obvious for one of ordinary skill in the art at the time the invention was made to enhance the desire for version checking as shown by Bloch so that using the browser engine for automatically checking the text files for the latest upgraded file version because this would enable the executing environment to be provided with the appropriate files according to the version expected for such environment as addressed above.

As per claim 2, Bloch discloses XML format (Fig. 1, 2, 4,5).

As per claim 3, Bloch discloses server central source for managing and distributing applications or modifications for applications (e.g. *upgrades, fixes* - pg. 3, para 0032; pg. 5, para 0045-0046; pg. 12, para 0109).

As per claim 4, refer to claim 3 and Bloch's Fig. 1, 2, 4,5.

As per claim 5, Bloch discloses executing an application, sending a request and executing the application in parallel while waiting for response from the request (e.g. … *reports to the Application Handler 302*, … *periodically updates* -- pg. 9, para 0080 – 0082 – Note: resolving a URL with data retrieval while leaving the GUI window on for being updated on tree events changes and notified of download status is equivalent to executing application while waiting for remote response)

As per claim 6, Bloch discloses connectionless application execution (e.g. pg. 8, para 0069; pg. 12, para 0108)

As per claim 8, Bloch discloses modifying application by using a newer text files replacing older files (*upgrades, fixes* - pg. 3, para 0032; *most recent …version* - pg. 12, para 0107).

As per claim 9, Bloch discloses graphical user interface (e.g. Fig. 6).

As per claim 10, Bloch discloses application being communication preferences for database invocation (e.g. pg. 7, para 0063; Preference Handler 303 - Fig. 4)

As per claim 11, Bloch discloses data management application (e.g. step *52* – Fig. 5; Manager 301 -Fig. 4 - Note: downloading files to assemble manager module reads on application being a management application).

As per claim 12, Bloch discloses component being part of logic of application (pg. 1, para 0012; pg. 4, para 0037).

As per claim 25, Bloch discloses a computer-readable medium having program code on a computer system to perform a method comprising:

installing a plurality of text files, each defining a component of the application (e.g. Fig. 1; steps 66, 68, 70 - Fig. 5; Fig. 6; pg. 10-11, para 0095, 0096);

installing a program wherein said program comprises instructions for using a combination of said text files to create an application (e.g. *AVM 221* - Fig. 2; para 0037, pg. 4); and

creating said application on said client system according to said program (e.g. Fig. 2, 5);

wherein said application is executable independent from said program (e.g. *Client Tasks, Host tasks* – para 0086-87, pg. 10; *once ... initialized ... user interact ...visible frame ... next user* interaction ... *System handler 315 to exit* - para 00100, pg. 11; *message box ... tasks ... unique to the operating system* – para 0102, pg. 11; *remote procedures* – para 0103, pg. 11; *Database handler ... user to access data ... Database* – para 0105, pg. 12; *user interface ... accessing any database ... to test new software* – para 0109 – Note: user-driven actions - para 0057-0058, pg. 6 - to invoke methods, system tasks or script based remote calls or for testing of components **read on** executables of assembled applications within the AVM non-application-specific framework – see para 0037, pg. 4(emphasis added) – and/or execution thereof that depend **on user**, i.e. **contextually independent** from the Gui layer or interface support by the launched AVM).

receiving automatically any updated versions of said execution environment files (e.g. *to make sure ... most current versions ... download and install automatically*-- para 0051-0053, pg. 5-6) in response to version checking.

But Bloch does not explicitly disclose automatically receiving any updated versions of said text files in response to version checking. Bloch teaches database storage for download support for version upgrades and the implicit version checking by browsers of markup files as set forth in claim 1. Thus, it would have been obvious for one of ordinary skill in the art at the time the invention was made to enhance Bloch deployment and XML processing environment so that not only script or virtual machine files, but also the text files such as the XML files are checked for update and automatic re-download, according to the version checking as known in browser technologies set forth in claim 1, because of the desirability to conform not only application files but also specification files, a concept inherent to browsers using XML metadata without which format and version conformance would potentially create application execution conflicts.

4. Claims 13-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bloch et al., USPN: 2002/0129129 (hereinafter Bloch), in view of Landsman et al., USPN: 6,314,451 (hereinafter Landsman).

As per claim 13, Bloch discloses a computer system with bus, processor coupled to a bus (*Client PC* - Fig. 1; pg. 12, para 0108) for implementing an application comprising the steps:

receiving (text files);

executing (program resident);

creating (application), wherein said application is executable independent from said program (refer to claims 1, 25). All these steps limitations have been addressed in claim 1; hence are rejected herein with the corresponding rejections as set forth therein, respectively.

But Bloch does not disclose uploading results from using said application to a server computer system. However, Bloch teaches legacy browsers (para 0048) so as to obviate redeploying using alternate means as well as remote persisting of records on reuseable user application data until the user decide to change the application preferences (para 0069-0070, pg. 8). Landsman teaches a browser environment where the user can customize application-related preferences by providing mouse-clicking interface analogous to the user-driven method of Bloch. Retransmitting of deployed application results are evidenced further in Landsman's method wherein logs of application execution data are uploaded to a server (e.g. col. 31, line 62-67). Based on the desirability to persist user preferences and the implied benefits of legacy of schema being used for users as mentioned above, it would have been obvious for one of ordinary skill in the art at the time the invention was made to enhance Bloch server database records so that there is an uploading of application results as taught by Landsman so that improvement of previous results or user schema preferences would lend some insight as implied via Bloch's teachings or via the analysis of logs data by Landsman.

As per claim 19, Bloch discloses text files particular to client system (e.g. pg. 4, para 0037; pg. 5, para 0047, 0050)

As per claims 14-18, 20-24, these claims correspond to claims 2-6, 8-12 respectively, hence are rejected with the corresponding rejections as set forth therein, respectively.

Response to Arguments

5. Applicant's arguments filed 11/01/07 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

USC § 103 Rejection:

(A) Applicants have submitted that Bloch's application is "executed by the AVM 221 on the client device 10"; and this is opposite to what is now claimed as 'application that is executable independent from the program that created the application' (Appl. Rmrks pg. 7-8). In terms of assessing merits to what is termed as 'executable independent from the program', there is not implementation details in the Claim or anywhere in the Specifications that enforces a particular type of execution independency between a program being responsible for assembling client executables and the executing per se of said executables. Scanning the Specifications, the *application* being referred to as being *executable* amounts to those executables that can be executed at the client environment so that they do not require communication with a ongoing session with a server or depend on a online timeliness. The disclosure describes at length about an agent for assembling applications that are executables by a client computer (Specifications pg. 17) based on components that define a GUI derived from content of XML files that define structural relationship between data related to communications that would otherwise be needed with a server live session. The Specifications stops short at just describing XML files being instrumental for assembling (via an agent intervention) applications residing and executable on the client (Specs pg. 21). There is no enablement type of details as to show such assembled application when executed would be strictly detached from the runtime of either the agent or the framework that downloads files. As for the merits of the above phrase ('application that is executable independent from the program that created the application') it is therefore reasonable to conclude that the assembled applications as claimed will always execute in a context of their own at the client end, and such runtime context is not dynamically linked in anyway to the memory runtime context of framework instance that originally provided the GUI components to download files and assemble code components (see Specifications Fig. 2, 3, 4), as these are customized based on a end-user (see Specs, pg. 22, bottom). Curiously enough, Bloch provides the same application user-based invocations happening within and based on instance of a framework being launched earlier to support creation of executable components. That is, Bloch's framework launches an application-non-specific core AVM instance of the AVM tool, and by linking server data with the layout of the tool, enables GUI components to support file-download and customization - by a developer - of methods or applications components that are then executable or launched by an user. Such invoking by the user (event that depend solely on user) thus entails execution which is deemed independent from the AVM core program that has been initialized to set up the GUI layout and of the non-application-specific OS neutral layer of Bloch's framework - framework purported for supporting creation of components assembled from downloaded XML files based on the client specific OS. The rejection has provided sufficient citations to demonstrate that the client-driven applications are executed independent from the original AVM core program that has been launched to set a GUI context support; i.e. the runtime context of the AVM being started from the outset very remotely dissociated from the runtime events like user-based invocations of executables --resulting from the XML files support – like script remote procedures, database queries, test of program, executing OS tasks or subsystem low layer calls. The argument is deemed not sufficient to overcome the rejection, and the phrase as above recited does not provide sufficient teaching as to preclude Bloch's invocations of executable from being really dependent from the instantiation of the core AVM instance, as proffered at length via citations support in the rejection. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference(s).

(B) In all, the claims will stand rejected as set forth in the Office Action.

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Tuan Anh Vu

Tuan A Vu
Patent Examiner,
Art Unit 2193
December 11, 2007